

3. Izjave in odločitve, pogojni izraz, pogojni stavek

3.1. Izjave

V logiki je izjava trditev, ki je lahko *resnična* ali *neresnična*. Izjave povezujemo z izjavnimi vezniki v sestavljene izjave. Pri tem je vrednost (resničnost oziroma neresničnost) sestavljene izjave odvisna od vrednosti posameh komponent ter logičnih veznikov. Spomnimo se logičnih veznikov in resničnostnih tabel.

Zgled: Najbolj znani logični vezniki.

\neg negacija

\wedge konjunkcija

\vee disjunkcija

\Rightarrow implikacija

\Leftrightarrow ekvivalenca

$\underline{\vee}$ ekskluzivna disjunkcija

$\overline{\wedge}$ nand

$\overline{\vee}$ nor.

Vrednost "res" včasih zapišemo kot "1", "ni res" pa kot "0". V tem primeru lahko za vsako sestavljeno izjavo zapišemo njeno **resničnostno tabelo**.

3.2. Zgled: Določite resničnostno tabelo izjave: $(\neg p \vee (q \wedge p))$

```
p q I ( $\neg p \vee (q \wedge p)$ )
=====
0 0 I 1
0 1 I 1
1 0 I 0
1 1 I 1
=====
```

V Pythonu je logična vrednost "res" zapisana kot `True`, vrednost "ni res" pa kot `False`. Resničnostna tabela v tem primeru zglada takole:

```
p q I ( $\neg p \vee (q \wedge p)$ )
=====
False False I True
False True I True
True False I False
True True I True
=====
```

3.3. Predikati in kvantifikatorji

Predikat $P(x)$, $Q(x,y)$, ..., je poljubna funkcija, ki ima zalogo vrednosti $\{\text{True}, \text{False}\}$. Poznamo (vsaj) dva kvantifikatorja:

$\forall x: P(x)$... Za vsak x velja $P(x)$.

$\exists x: P(x)$... Obstaja tak x , da velja $P(x)$.

O predikatih in kvantifikatorjih bomo več povedali kasneje.

3.4. Pogoji

Pogoj v jeziku Python je lahko poljuben izraz. Pravimo, da pogoj ni izpolnjen, če ima vrednost `False`, `None`, `0`, prazen niz, prazen seznam, ... V vseh ostalih primerih je pogoj izpolnjen.

3.5. Primerjalni operatorji:

Primerjalni operatorj je dvomestni predikat $Q(x,y)$, ki pa je zapisan v infiksni obliki:

$x Q y$.

- $x < y$ (x je manjši od y)
- $x > y$ (x je večji od y)
- $x <= y$ (x je manjši ali enak y)
- $x >= y$ (x je večji ali enak y)
- $x == y$ (x je enak y)
- $x != y$ (x je različen od y)
- $x \text{ is } y$ (x je isti kot y)
- $x \text{ is not } y$ (x ni isti kot y)
- $x \text{ in } y$ (x je vsebovan v y)
- $x \text{ not in } y$ (x ni vsebovan v y)

Rezultat primerjave je vedno logična vrednost **True** ali **False**. Posamezne pogoje lahko sestavljamo v bolj zapletene pogoje z uporabo logičnih operatorjev:

- $a \text{ and } b$ (logični in: \wedge)
- $a \text{ or } b$ (logični ali: \vee)
- $\text{not } a$ (logična negacija: \neg)

Operator `not` vrača logično vrednost `True` ali `False`, medtem ko operatorja `and` in `or` najprej izračunata vrednost prvega operanda. Če je izpolnjenost celotnega izraza s tem že dolžena, vrneto njegovo vrednost, sicer pa izračunata še vrednost drugega operanda in nato izračunata in vrneto vrednost celotnega izraza.

3.5. Zgled:

Ker so negacija, konjunkcija in disjunkcija poln nabor logičnih veznikov, je mogoče z njimi izraziti poljubno izjavo. Zapišite izraze $p \Rightarrow q$, $p \Leftrightarrow q$, $p \bar{\wedge} q$, $p \bar{\vee} q$ v Pythonu (vezniki so implikacija, ekvivalenca, `nand` in `nor`).

$p \Rightarrow q$: `not p or q`

$p \Leftrightarrow q$: `(not p or q) and (not q or p)`

$p \bar{\wedge} q$: `not (p and q)`

$p \bar{\vee} q$: `not (p or q)`

Primerjave lahko tudi verižimo: $x < y < z$ pomeni isto kot $x < y$ and $y < z$

3.6. Naloga: Ugotovite, ali je dano leto prestopno, tako za število v spremenljivki `leto` izračunate izraz `prestopno`, ki je bodisi `True` ali `False`.

```
prestopno = leto % 4 == 0 and leto % 100 != 0 or leto % 400 == 0
```

3.7. Pogojni izraz

Pogojni izraz v jeziku Python vrne vrednost a, če je pogoj p izpolnjen, sicer pa vrne vrednost b.

- a if p else b

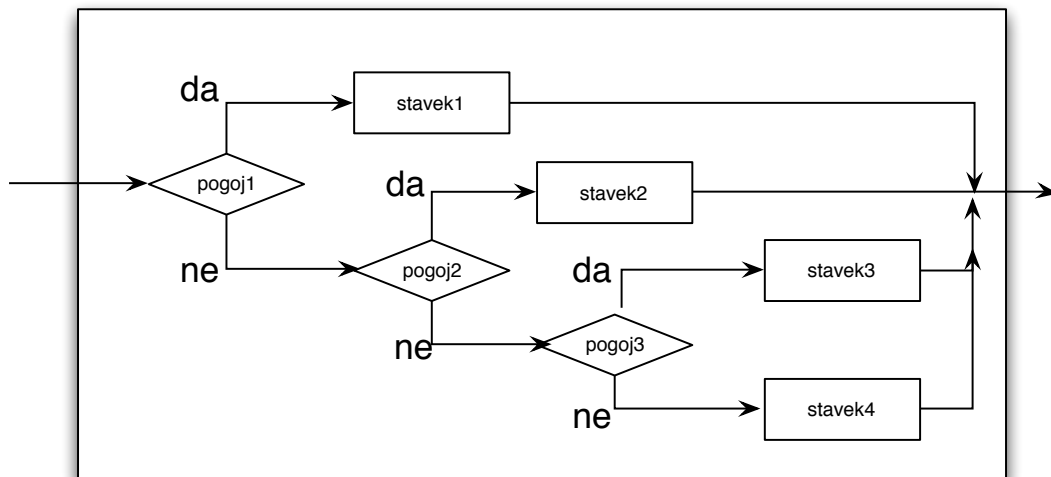
Nadaljevanje prejšnjega primera: Izračunajte število dni v letu leto. Dodamo naslednji stavek:

```
steviloDni = 366 if prestopno else 365
```

3.8. Pogojni stavek

Pogojni stavek nam omogoča, da enega ali več stavkov izvršimo samo, če so izpolnjeni vsi potrebni pogoji.

```
if pogoj1:  
    stavek1  
elif pogoj2:  
    stavek2  
elif pogoj3:  
    stavek3  
else:  
    stavek4
```



Pogojni stavek z dvema elif in else. Npr. stavek3 se izvede, če je resničen izraz:
(not pogoj1 and not pogoj2 and pogoj3)

Pogojni stavek je sestavljen iz obveznega dela if, poljubnega števila (lahko tudi 0) delov elif in največ enega dela else. V zgornjem primeru se bo izvedel natanko eden izmed štirih stavkov, kateri bo to, pa je odvisno od izpolnjevanja pogojev. Če je izpolnjen pogoj1, se bo izvedel stavek1 (in noben drug). Če pogoj1 ni izpolnjen, se preveri pogoj2. Če je ta izpolnjen, se izvede stavek2. Če tudi pogoj2 ni izpolnjen,

se preveri pogoj3. Če je ta izpolnjen, se izvede stavek3, sicer (če ni bil izpolnjen noben pogoj) pa se izvede stavek4.

Stavki morajo biti bolj odmaknjeni od roba, kot sam pogojni stavek.

Pri vsakem od delov lahko napišemo tudi po več stavkov skupaj.

```
if pogoj:
    stavek1
    stavek2
    stavek3
else:
    stavek4
    stavek5
```

Če je treba izvesti samo en stavek, ga lahko napišemo takoj za dvopičjem, vendar moramo paziti, da s tem ne otežimo branja programa (program naj bo vedno napisan čimbolj pregledno).

```
if pogoj1: stavek1
elif pogoj2:
    stavek2
    stavek3
else: stavek4
```

Zgled: Analiza ničel kvadratne funkcije. Naj bo $f(x) = a x^2 + b x + c$. Koliko ničel in kakšne ničle ima enačba $f(x) = 0$?

```
print("Analiza kvadratne funkcije f(x) = a*x**2 + b*x + c")
print("a,b, in c naj bodo cela števila.")
a = int(input("a = "))
b = int(input("b = "))
c = int(input("c = "))
if a == 0:
    print("Enačba je linearna.")
    if b == 0:
        if c == 0:
            print("Rešitev je cela realna os.")
        else:
            print("Rešitev je prazna množica.")
    else:
        print("Rešitev linearne enačbe je ",-c/b)
else:
    D = b**2 - 4*a*c
    if D < 0:
        print("Konjugirano kompleksna korena")
    elif D > 0:
        print("Realna korena.")
    else:
        print("Dvojni realni koren.")
```

3.9. Naloge

V vsaki od naslednjih nalog podatke preberemo s stavki `input`. Pri tem prebrani niz preoblikujemo v realno število z ukazom `float`.

1. Sestavite program, ki bo izračunal za dani x vrednost periodične realne funkcije, ki je na polodprtem intervalu $[0,10)$ definirana takole:
 - če je $0 \leq x < 2$, potem je $f(x) = x + x^2$
 - če je $2 \leq x \leq 4$, potem je $f(x) = |x-3|$
 - če je $4 < x < 7$, potem je $f(x) = (x - 5)^2$
 - če je $7 \leq x < 10$, potem je $f(x) = 12 - 5x$
2. Pravokotnik v ravnini, ki ima stranice vzporedne s koordinatnima osema, predstavimo s koordinatami dveh nasprotnih si oglišč: x_0, y_0 in x_1, y_1 . Sestavite del programa, ki bo za dani pravokotnik in dano točko (x, y) v ravnini preverila, kje leži točka: znotraj, zunaj ali na robu pravokotnika.
3. Sestavite del programa, ki bo za dani premici $y = ax + b$ ter $y = Ax + B$ v ravnini preverila, ali sta enaki, vzporedni, ali pa se sečeta v natanko eni točki (to točko naj tudi izračuna). Vsaka od premic je podana s smernim koeficientom in prostim členom: (a, b) in (A, B) .
4. Sestavite del programa, ki reši linearno enačbo: $a \cdot x + b = 0$. pri znanih a in b .
5. Sestavite del programa, ki izračuna ploščino trikotnika z oglišči v točkah $A(a, b)$, $B(c, d)$ in $C(e, f)$.
6. Sestavite program, ki ugotovi, ali so točke A, B in C iz prejšnjega primera kolinearne.
7. Sestavite program, ki ugotovi, ali si točke A, B in C sledijo v nasprotni smeri urinega kazalca.